

构建软件测试领域不确定性知识图谱

罗浩榕,朱卫星*,史涯晴*,万进勇

(陆军工程大学 指挥控制工程学院,江苏 南京 210001)

摘要:由于软件测试技术的不足和成本的限制,从历史软件测试资产中获取的可复用知识对象具有不确定性。为了更有效地复用历史测试知识,软件测试知识复用系统应当建模和评估知识的不确定性。模糊知识图谱是利用不确定性理论对典型知识图谱的一种扩展,在利用语义网络描述知识实体及其关系的基础上,通过模糊关系建模知识不确定性。通过分析软件测试资产,确定测试需求、测试用例和问题报告三类知识的表现形式和内涵,建立软件测试资产本体和领域特征本体作为知识图谱模式层,提出一种基于信息论的领域特征抽取方法,构建领域特征图谱。分析软件测试不确定性的产生原因,基于粗糙集理论对本体进行扩展,在软件测试资产和领域特征之间建立模糊关系。提出一种基于知识粒度的模糊置信度计算方法,用于构建软件测试模糊知识图谱,为软件测试中不确定性知识的复用提供了基础。通过一个指挥信息系统领域实例验证了方法的可行性。

关键词:软件测试;知识图谱;不确定性知识;粗糙集;知识粒度

中图分类号:TP311

文献标识码:A

文章编号:1673-629X(2022)07-0111-06

doi:10.3969/j.issn.1673-629X.2022.07.019

Construction of Knowledge Graph of Software Test Asset for Uncertain Knowledge

LUO Hao-rong, ZHU Wei-xin*, SHI Ya-qing*, WAN Jin-yong

(School of Command & Control Engineering, Army Engineering University of PLA, Nanjing 210001, China)

Abstract: Due to the limitation of software testing technology and cost, the reusable knowledge objects obtained from historical software testing assets have uncertainties. To reuse historical testing knowledge more effectively, the software testing knowledge reuse system should model and evaluate these uncertainties. Fuzzy knowledge graph is an extension of typical knowledge graph based on uncertainty theory, which models knowledge uncertainty by fuzzy relationship on the basis of describing knowledge entities and their relationships by semantic network. By analyzing software test assets, the representations and content of test requirements, test cases and problem reports were determined, and the software test asset ontology and domain feature ontology were established as the pattern layer of knowledge graph. A domain feature extraction method based on information theory was proposed to construct domain feature graph. By analyzing the causes of uncertainty in software testing, the software test assets ontology is extended with rough set theory, and the fuzzy relationship between software testing assets and domain features is established. A fuzzy confidence calculation method based on knowledge granularity is proposed to construct fuzzy knowledge graph of software testing, which provides a basis for the reuse of uncertain knowledge in software testing. The feasibility of the method is verified by an example in the field of command information system.

Key words: software test; knowledge graph; uncertainty knowledge; rough set; knowledge granularity

0 引言

软件测试通过动态和静态的测试技术发现软件缺陷,是软件质量保证的关键活动。在实际软件测试过程中,由于软件测试分析、设计和执行技术的不足,测试用例几乎不可能做到输入域的全部覆盖,因此软件

测试的结果并不总是处于可预知的范围内,从历史软件测试资产中获取的软件测试知识具有一定的不确定性。因此,建立软件测试复用系统必须考虑如何表达不确定性知识和求解不确定性问题,尽可能消除测试不确定性。传统的软件测试复用通常基于1991年提

收稿日期:2021-07-24

修回日期:2021-11-25

基金项目:国家重点研发计划项目(2018YFB1403400)

作者简介:罗浩榕(1996-),男,硕士研究生,研究方向为需求工程与智能软件测试;通讯作者:朱卫星(1978-),男,博士,副教授,硕导,研究方向为网络结构、云计算、大数据、军事需求与软件工程;通讯作者:史涯晴(1981-),女,博士,副教授,硕导,研究方向为智能测试、时空数据处理。

出的软件复用 3C 模型^[1],即被复用工件的概念 (concept)、内容 (content)、上下文 (context)。由于时间和成本的限制,这三个维度的确定性假设可能很难成立,精确且完整的描述软件测试的概念、内容和上下文是非常困难的^[2]。2012 谷歌提出了知识图谱的概念,对现实世界中的实体、概念、属性及其关系进行建模,与传统的专家知识系统相比有更强的语义表达能力和建模灵活性,在知识存储、检索和推理方面有更大优势。建立专业领域知识图谱通常采用“自上而下”的方式,根据建模需求和业务领域构建领域本体作为知识图谱的模式层,再通过业务数据对本体实例化。为了表达知识的不确定性,模糊知识图谱结合不确定性建模方法扩展经典本体^[3-4],建立不确定性知识的表示和推理机制,在软件测试中不确定性建模方法包括贝叶斯信念网络、粗糙集、模糊逻辑、隐马尔可夫模型等^[5]。其中粗糙集可以表示不确定性信息空间中的多属性决策函数,而软件测试过程需要考虑多种属性是典型的多属性决策过程,因此考虑基于粗糙集扩展软件测试资产本体。

1 软件测试中的不确定性

2014 年,在 SWEBOK 3.0 发布的软件工程知识体系中,将软件测试定义为“从一个通常是无限的执行域(集合)中选择合适的、有限的测试用例,对程序所期望的行为进行动态验证的活动过程”。因此用三元组 $\langle I, S, O \rangle$ 表示软件测试用例 T ,其中 I 是被测件输入域, O 是被测件输出域, S 是被测件的起始状态,另有函数 f 表示输入域到输出域的映射。那么,对于测试用例任一测试输入都有一个在已知输出域的结果,即预期结果:

$$T\langle I, S, O \rangle: [\forall i \in I \rightarrow \exists f(i) = o \wedge o \in O] \quad (1)$$

当测试结果和缺陷分布未知时,认为软件测试存在不确定性,即对于给定的输入和起始状态,可能出现在已知输出域之外的结果。

$$\text{Uncertainty } T\langle I, S, O \rangle: [\exists i \in I \rightarrow \exists f(i) = r \wedge r \notin O] \quad (2)$$

造成软件测试中存在不确定性的主要原因有^[5-6]:

- 在测试需求分析阶段,由于软件开发阶段,需求建模与真实的用户需求和相应问题领域存在一定差异,软件测试继承了软件需求的差异性。

- 测试设计阶段,由于经济成本和时间成本的限制,遍历输入域的详尽测试无法实现。因此,对于每一个被测件都必须选择有限的测试用例子集,这可能无法检测所有的缺陷,造成了一定的不确定性。

- 对于许多软件系统例如嵌入式软件,完整地复现用户场景并不现实,因此软件测试通常在模拟测试环境中实施,测试环境与真实环境的差异也引入了不确定性。

2 软件测试模糊知识图谱

具有较大复用价值的软件测试资产主要包括三类:测试需求、测试用例、问题报告。这三类测试资产涵盖了软件测试分析、设计、执行、质量估计的全生命周期,是软件测试知识的集中体现。首先依据本体从三类测试资产中提取软件测试知识实体,形成测试资产知识图谱,结合行业标准中的领域知识形成领域术语图谱,再建立测试资产与领域特征的关联关系,采用基于知识粒度的方法为资产-特征赋予关系置信度。该文面向软件测试资产建立不确定性知识图谱的总体框架如图 1 所示。

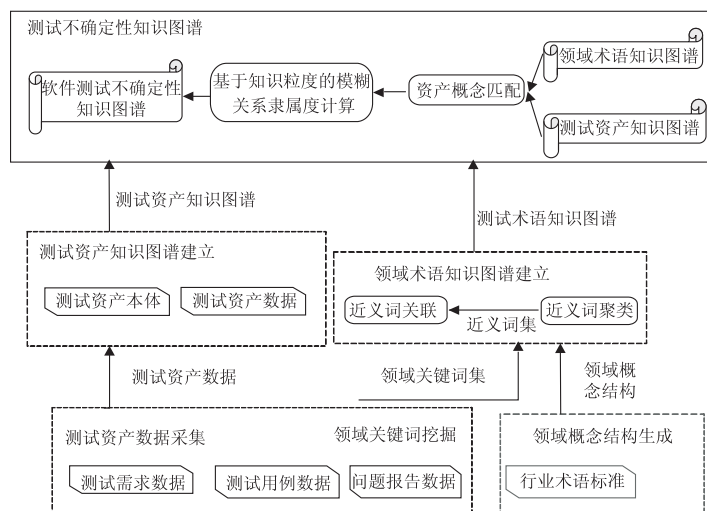


图 1 软件测试资产不确定性知识图谱建立总体框架

基于上述总体框架,对测试资产本体、领域术语图谱构建方法、模糊关系置信度计算 3 个方面进行阐述。

2.1 测试资产本体

软件测试资产中包含大量信息,这些信息可能通过正确的表示和处理转化为适用的知识,应用知识管理方法和原则管理测试资产,可以促进测试过程中的复用^[7-8]。软件测试资产主要有三个对象类,即测试需求、测试用例和问题报告。

定义1:测试需求。测试需求确定测试目标、测试内容和质量特性,确定测试充分性要求,提出测试活动应当遵循的基本原则和要求,确定测试资源和技术需求。测试需求(Test Requirement, TR)包括七元组:测试目标(Target, Ta)、被测件描述(Description, De)、测试阶段(Phase, Ph)、测试类型(Type, Ty)、测试环境(Environment, En)、测试方法(Method, Me)、判断规则(Evaluate, Ev),记为:

$$TR = \langle Ta, De, Ph, Ty, En, Me, Ev \rangle \quad (3)$$

经验表明,同一领域的测试项目往往遵循相同的行业规范、标准等,同一类型软件的使用场景、功能结构、性能要求等存在很大相似性,因此具备了知识复用的基础。

定义2:测试用例。测试用例是指测试人员根据软件需求规格说明和程序内部结构,精心设计出的一组测试输入、执行条件以及预期结果,用于核实软件是否满足某个特定需求。测试用例(Test Case, TC)包括五元组:需求追踪项(Required Trace, Rt)、前提条件(Precondition, Pr)、输入数据(Input Data, In)、执行步骤(Operation Step, Ops)、预期结果(Except Result, Er),记为:

$$TC = \langle Rt, Pr, In, Ops, Er \rangle \quad (4)$$

测试用例设计是软件测试的核心工作,对测试用例的复用是目前研究最多的,除了查找相似的测试用例直接获取历史经验,还可以归纳形成测试设计模式。

定义3:问题报告。问题报告是对未通过的测试用例进行记录和总结,提出复现软件缺陷的最短路径,分析定位错误。问题报告(Bug Report, BR)包括五元组:用例追踪项(Case Trace, Ct)、实际结果(Actual Result, Ar)、缺陷特征(Bug Character, Bc)、缺陷定位(Bug Location, Bl)、严重程度(Bug Severity, Bs),记为:

$$BR = \langle Ct, Ar, Bc, Bl, Bs \rangle \quad (5)$$

问题报告中蕴含了测试人员对软件缺陷的总结,可以获取软件缺陷模式知识帮助后续开发人员定位缺陷,对于修复问题具有重要意义。从测试资产原始数据中获取以上三类对象的实例,每条实例作为知识图谱中的一个节点,这些节点按照固有的继承和关联关系相连接,这部分关系由原数据直接得到,不赋予模糊置信度。

2.2 领域术语图谱

软件测试的决策因素主要包括被测件本身的特征、行业特征和软件测试本身特征。因此,将软件资产中的概念分为三类 $\langle W_d, W_t, W_s \rangle$,其中行业概念 W_d 指被测件应用的问题领域相关概念;软件测试概念 W_t 指软件测试本身特有概念词汇;软件概念 W_s 指描述被测软件的特征概念,包括软件架构、接口协议、编程语言等,每一类概念包括若干棵领域术语树。

领域术语树中每个节点代表一条领域术语,节点间存在两类关系:子术语关系、近似术语关系。

3 领域术语图谱构建

领域术语图谱构建过程如图2所示。

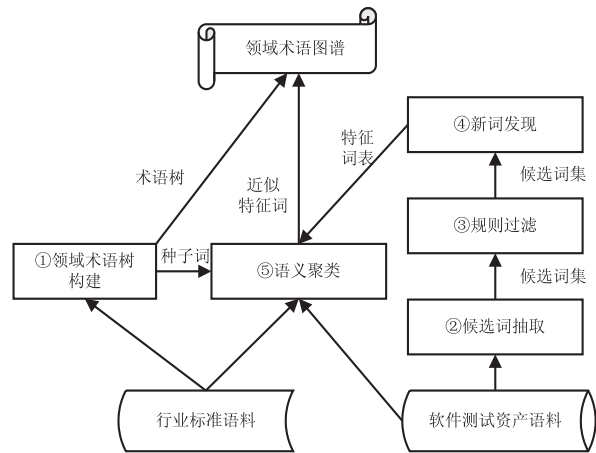


图2 领域术语图谱构建过程

①基本术语树构建,基于行业标准等权威数据建立领域概念结构。

根据国标、国军标等行业标准中关于行业领域、软件测试、软件系统的相关定义,构建层次化的术语树。

②候选词提取,基于互信息和左右熵从测试资产语料中发现候选词集^[9]。

行业领域的术语词往往与一般词汇不同,通用分词工具的结果可能遗漏一些固有搭配,例如,指挥信息系统可能被分解为三个词:指挥、信息、系统。互信息和左右熵分别计算词的内部凝和度和边界自由度。

词语 x 和词语 y 的互信息 $MI(x, y)$ 体现了两个词之间的依赖程度,其计算方法如下:

$$MI(x, y) = \log_2 \frac{P(x, y)}{P(x)P(y)} \quad (6)$$

其中, $P(x, y)$ 是词语 x 、 y 的共现词频, $P(x)$ 、 $P(y)$ 定义类似。互信息越高,说明词语 x 、 y 相关性越高,组成术语短语的可能性越大。

词语 x 的左右熵 $H(x)$ 指该词左右边界的信息熵,其计算方法如下:

$$H(x) = H_L(x) + H_R(x) = \left(- \sum_{a \in A} P(ax | x) \cdot \right.$$

$$\log_2 P(ax|x)) + (- \sum_{\forall b \in B} (xb|x) \cdot \log_2 P(xb|x)) \quad (7)$$

其中, A 是 x 的前序词语集, B 是 x 的后序词语集。 x 的左右熵越大, 说明该词的周边词越丰富, 自由程度越高, x 独立成为一个词的可能性越大。

通过计算互信息和左右熵进一步处理通用分词工具的分词结果, 可以发现行业领域的特有词语搭配, 这些特有搭配往往就是行业术语。

③规则过滤, 筛选去除非词。

由于语料中噪声和算法的误差, 前述无监督的候选词抽取结果中可能含有少量的非词, 需要通过专家规则进行筛选。

④新词发现, 对比软件测试资产语料和背景语料, 进一步筛选特征词^[10]。

前述通过词频提取的候选词集中除了特有的术语还包括通用的词汇和短语。因此, 对于候选词 x , 通过对比该词在不同语料中的信息量进行打分, 其打分函数定义如下:

$$S(x) = \log_2 P_D(x) - \frac{1}{n} \sum_{i=1}^n (\log_2 P_{T_i}(x) + \mu) \quad (8)$$

其中, 平滑参数 μ 是一个较小的正数, 为了避免在背景语料中 x 词频为 0 的情况下得分为无穷大。在该打分函数中, 如果词语 x 在软件测试资产语料中词频 $P_D(x)$ 比在背景语料集中语料 T_i 的词频 $P_{T_i}(x)$ 更高, 那么会得到更高的分数, 选择得分前 50% 的候选词作为特征词。

⑤语义聚类, 通过词向量聚类获取近似领域词集^[11]。

结合特征词表, 在软件测试资产和行业标准语料进行分词并训练一个 Word2Vec 模型, 根据 Word2Vec 得到的词向量来对词进行聚类。算法以术语树中的节点作为种子词, 利用相似的传递性计算近似词集。为了尽量避免迭代过程中的指数偏移, 以指数衰减的方式迭代计算词向量余弦相似度, 将近似词聚为一类, 每一类包含且仅包含一个种子词, 连接到基本术语树中的相应节点形成完整的领域术语图谱。相似度计算方法如下:

$$N_0(x, k) = \frac{v_x \cdot v_k}{\|v_x\| \|v_k\|} \quad (9)$$

$$N_t(x, y) = \frac{v_x \cdot v_y}{\|v_x\| \|v_y\|} e^{-\alpha t} \quad (10)$$

参数 $\alpha > 0$ 是指数衰减常数, 决定相似度随迭代的衰减速度。第一轮迭代时, 将所有与种子词 k 相似度 $N_0(x, k)$ 超过阈值的特征词 x 加入集合; 之后每轮迭代将与集合中特征词 x 相似度 $N_t(x, y)$ 超过阈值的特征词 y 加入集合; 直到集合元素不再增加时, 得到 x

的近似词集合, 将该集合所有元素加入术语树, 并与种子词 k 建立相似关系, 最终得到领域术语图谱。

4 基于知识粒度的模糊置信度计算

经典模糊知识系统中, 模糊置信度由专家根据先验知识赋予, 这种方法较为主观, 且在实例较多的知识系统中工作量十分庞大。粗糙集理论是一种处理不精确、不完全数据的数学理论, 目前已经广泛应用于机器学习、知识发现、数据挖掘等场景。在经典 Pawlak 粗糙集理论中, 知识指根据特征对论域进行分类所得的子集, 知识库定义为一个粗糙近似空间 $AS = (U, R)$, 其中 U 是论域, R 是在 U 上的一族等价关系。

定义 4: 知识粒度^[12]。设 $AS = (U, \mathfrak{R})$ 是一个知识库, $R \in \mathfrak{R}$ 是一个等价关系, R 对 U 划分的等价类集 $U/R = \{X_1, X_2, \dots, X_n\}$, 那么知识 R 的知识粒度为:

$$GD(R) = \frac{|R|}{|U|^2} = \frac{\sum_{i=1}^n |X_i|^2}{|U|^2} \quad (11)$$

知识粒度可以描述知识的分类能力, 知识粒度越小, 其划分的等价类越精细, 那么知识的分类能力越强。软件测试知识的不确定性可以归为两类, 一是概率性, 即软件测试知识本身具有概率性; 二是模糊性, 即在知识图谱中用于描述知识的术语不清晰或不准确。以模糊置信度建模特征术语对软件测试资产的影响程度。每一条软件测试资产数据都涉及若干特征术语, 这些特征术语体现了该软件测试资产的决策因素, 将知识的概率性和模糊性统一以置信度的形式表示。置信度越高表示特征术语 w 对测试资产 A 的代表性越强。文中特征术语和软件测试资产的模糊置信度实质上即通过特征术语区分软件测试资产的能力^[13-14], 有研究表明决策属性的知识粒度越小则其信息熵越大^[15-16], 基于知识粒度计算模糊置信度, 评价相应特征对软件测试资产的信息熵影响, 也即其决策影响程度是有效可行的。对于每一条资产 A , 通过三类概念特征集合 $\{w_d, w_t, w_s\}$ 表征, 这里 w_d 表示资产 A 中涉及的问题领域概念子集 $w_d \in W_d$, w_t, w_s 类似。在图谱中搜索与资产 A 相关的所有概念, 相似概念则合并为一项, 通过概念特征集 $\{w_1, w_2, \dots, w_n\}$ 表示一条资产数据。计算术语 w_i 作为资产 A 特征的模糊置信度分为两步: 粗糙近似空间构造、模糊置信度计算。

4.1 粗糙近似空间构造

构造粗糙近似空间 $AS = (U, \mathfrak{R})$, 即确定论域 U 和等价关系族 \mathfrak{R} , 由于 \mathfrak{R} 中的等价关系实际上指特征属性相同或近似而不可区分的论域对象集。因此构造 Pawlak 粗糙近似空间等价于确定一个知识表达系统 $S = (U, P, V, f)$, 其中 U 为对象的非空有限集合, P 是

属性的非空有限集合, V 是 P 的值域, $f: U \times P \rightarrow V$ 是信息函数为每个对象的每个属性赋值, 有 $\forall a \in A, \forall x \in U, f(x, a) \in V_a$ 。文中特征术语相同的测试资产是不可区分的对象, 论域 U_A 为软件测试资产实例集, V 为论域中所有测试资产的特征术语全集, P 为 V 的父概念集。若以模糊知识图谱中的测试资产全集为论域, 存在两个问题: 一是搜索空间过大, 计算效率低; 二是大部分测试资产与资产 A 特征无关, 大量无关测试资产被分为一类使得知识粒度接近于 1, 不能有效区分特征术语 w_i 对资产 A 的分类能力。因此, 需要筛选论域, 构造与资产 A 和特征术语 w_i 粗糙近似空间, 在软件测试模糊知识图谱 G_F 中确定测试资产实例 A 与其特征术语 w_a 相关的粗糙近似空间方法如下:

算法 1: 软件特征粗糙近似空间构造。

Input: 软件测试模糊知识图谱 G_F 、测试资产实例 A 、 A 的一个特征术语 w_a

Output: 资产 A 相关的知识表达系统 S_A

#搜索特征 w_i 的中心特征 k 及其父概念 K

1 search k, K from G_F , which $w_a \approx k \wedge k \in K$

#搜索概念 K 相关的所有 A 同类测试资产集 U_A 作为

论域

2 for every $k \in K$:

3 search all A_i from G_F , which $A_i \sim k \wedge \text{type}(A_i) = \text{type}(A)$,

put A_i in U_A

#搜索条件属性集 P 及其值域 V , 信息函数 f

4 for A_i in U_A :

5 search k_{ij}, K_{ij} , which $A_{ij} \sim k_{ij} \wedge k_{ij} \in K_{ij}$

6 put K_{ij} in P

7 put $((A_{ij}, K_{ij}) \rightarrow k_{ij})$ in f

8 search $k \in K_{ij}$, put k in V

#输出粗糙近似空间

9 output: (U_A, P, V, f)

4.2 模糊关系置信度计算

通过概念类 K 的知识粒度刻画资产 A 对术语 w_i ($w_i \in K$) 的置信程度, K 的知识粒度越小即认为这一类概念对测试资产的划分描述能力越强, 是有效的分类特征^[15-16], 通过 z 型模糊器计算得出的 A 与 w_i 的模糊关系置信度越高。

定义 5: 概念 K 的决策能力。以概念 K 作为条件属性对论域做划分, K 的若干子概念 $\{w_{k1}, w_{k2}, \dots, w_{kl}\}$, 将论域划分为 l 个等价类。那么 K 的分类能力 C_K 计算如下:

$$C_K = \frac{\sum_{i=1}^l |w_{ki}|^2}{|U|^2} \quad (12)$$

定义 6: 模糊关系置信度。设有测试资产 A , w_i 是表征 A 的一个概念特征, 根据前述粗糙近似空间构造方法得到论域 U_A 和条件属性集 $P = \{K, K_1, \dots, K_m\}$, 那么 w_i 与 A 的模糊关系隶属度为:

$$\mu_{(A, w_i)} = \text{zmf}(C_K) = (1 + e^{\rho(C_K - \theta)})^{-1} \quad (13)$$

参数 ρ, θ 决定模糊函数的平滑程度。模糊关系置信度 $\mu_{(A, w_i)} \in (0, 1)$ 描述了在产生资产 A 的软件测试活动中, 术语 w_i 所代表的领域知识对决策过程的影响程度, 置信度越高则影响力越强, 在对资产 A 进行知识推理时, 该特征的重要性越高。

5 实例验证

本节通过一个实例验证了该方法的可行性

5.1 实验条件

采用 neo4j 图形数据库构建实例知识图谱, 数据来源于作者所在单位承担的软件测评任务和网络搜集的行业标准等数据, 具体见表 1。

表 1 实验条件

环境配置	测试资产来源	行业标准来源
AMD Ryzen 7 4800HS、	XXX 训练信息系统测试资产	GJB 3510A-2018
NVIDIA GeForce RTX 2060	XXX 后勤信息系统测试资产	GJB 2434-1995
neo4j-community-4.1.9	XXXX 指挥信息系统分系统测试资产……	GJB 5296-2004……

5.2 结果分析

软件测试资产知识图谱中知识实体、关系的情况如表 2 和表 3 所示。

表 2 知识实体统计

知识类型	实体数量	平均关系数量
测试需求	4 820	2.4
测试项	11 460	4.5
测试用例	13 620	1.3
软件缺陷	4 780	1.0
中心概念	819	2.9
资产特征	1 244	16.6

表 3 资产知识关系情况

关系名称	头实体	尾实体	数量
项目组成	测试项	测试项目	11 460
覆盖	测试项	软件需求	11 460
设计依据	测试用例	测试项	13 620
发现于	问题报告	测试用例	4 780
子需求	软件需求	软件需求	4 660
子项	测试项	测试项	11 019

从表 2、表 3 可以发现软件测试资产知识图谱实体关系较为稀疏, 知识实体多且分散, 如果用关系数据

库存知识实体,一方面不便于建立实体-实体、实体-特征的关联关系,进行复杂条件查询时大量链接操作会极大影响查询效率。另一方面,知识图谱比关系数据库更有利于后续知识推理任务。

领域术语图谱情况如表 4 所示。

从表 4 可以看出,由于测试人员撰写不规范,软件测试资产中的领域术语存在大量近似词,模糊化的知识表述对知识复用造成了严重阻碍,直接进行关键词查询可能造成错误或遗漏。领域术语图谱可以视为一种结构化的近义词典,通过先验知识提高了知识实体查询的准确率和查全率,大大减弱了测试知识复用中

由测试资产模糊性带来的知识错漏问题。

表 4 领域术语图谱实体关系

关系名称	头实体	尾实体	数量
子概念	中心概念	中心概念	727
近似	资产特征	中心概念	1 566
关键词	软件需求、测试项、 测试用例、问题报告	资产特征、 中心概念	20 732

表 5 对比了使用关键词查询测试知识实体时,有、无领域术语图谱辅助的两种情况。

表 5 准确率/召回率对比

关键词类别 查询方法	专业术语	功能名称	接口名称	简称缩写	平均
领域术语图谱辅助	0.82/0.87	0.75/0.76	0.81/0.82	0.72/0.76	0.776/0.798
直接查询	0.56/0.64	0.52/0.63	0.49/0.62	0.21/0.46	0.452/0.588

6 结束语

该文设计了一种可以表达不确定性知识的软件测试知识图谱构建框架,能够利用测试需求、测试用例、问题报告三类软件测试资产和行业标准等领域知识建立软件测试知识图谱,提出了一种基于信息量的领域术语挖掘方法,在此基础上构建了领域术语图谱作为软件测试资产的特征。为了强调软件测试中的不确定性,使用通过特征术语的知识粒度建模计算其在软件测试过程中的决策影响程度。通过为软件测试资产的特征赋予模糊置信度,构建了软件测试模糊知识图谱,为后续软件测试不确定知识推理做铺垫。

参考文献:

- [1] LATOUR L, WHEELER T, FRAKES B. Descriptive and predictive aspects of the 3Cs model: SETAI working group summary[C]//Proceedings of the first international symposium on environments and tools for Ada. New York: Association for Computing Machinery, 1991: 9-17.
- [2] CAPILLA R, GALLINA B, CETINA C, et al. Opportunities for software reuse in an uncertain world: from past to emerging trends[J]. Journal of Software Maintenance & Evolution Research & Practice, 2019, 31(8): e2217.1-e2217.16.
- [3] WANG S, ZHANG Y, LIAO Z. Building domain-specific knowledge graph for unmanned combat vehicle decision making under uncertainty[C]//2019 Chinese automation congress (CAC). Hangzhou: IEEE, 2019.
- [4] SUN P, GU L. Fuzzy knowledge graph system for artificial intelligence-based smart education[J]. Journal of Intelligent and Fuzzy Systems, 2020, 40(2): 1-12.
- [5] MOIZ S A. Trends in software testing: uncertainty in soft-

- ware testing[M]. Singapore: Springer, 2017: 67-87.
- [6] WOUFF D, GOLDSTEIN M, COOLEN F. Bayesian graphical models for software testing[J]. IEEE Transactions on Software Engineering, 2002, 28(5): 510-525.
- [7] ZHANG Nan, CHAI Haiyan, HAN Xinyu. Research on the reuse of test case for warship equipment software[C]//IEEE international conference on software quality. Lisbon: IEEE, 2018: 64-69.
- [8] DADKHAH M, ARABAN S, PAYDAR S. A systematic literature review on semantic web enabled software testing[J]. Journal of Systems and Software, 2019, 162: 110485.
- [9] 刘伟童, 刘培玉, 刘文锋, 等. 基于互信息和邻接熵的新词发现算法[J]. 计算机应用研究, 2019, 36(5): 1293-1296.
- [10] 孙珠婷, 顾倩颐. 领域概念术语提取中特征项自动抽取方法[J]. 计算机工程与设计, 2012, 33(8): 3189-3193.
- [11] GAGLIARDI I, ARTESE M T. Semantic unsupervised automatic keyphrases extraction by integrating word embedding with clustering methods[J]. Multimodal Technologies and Interaction, 2020, 4(2): 30.
- [12] 王树锋, 吴耿锋, 潘建国. 基于粗糙集的知识粒度及粒度关系研究[J]. 计算机工程与应用, 2007, 43(14): 38-41.
- [13] PANCERZ K. Semantic relationships and approximations of sets: an ontological graph based approach[C]//International conference on human system interaction. Sopot, Poland: IEEE, 2013.
- [14] UPADHYAYA S, ARORA A, JAIN R. Rough set theory: approach for similarity measure in cluster analysis[C]//Proceedings of the 2006 international conference on data mining. Las Vegas: CSREA Press, 2006: 353-356.
- [15] 苗夺谦, 王珏. 粗糙集理论中知识粗糙性与信息熵关系的讨论[J]. 模式识别与人工智能, 1998(1): 34-40.
- [16] 吕萍, 常玉慧, 钱进. 知识粒度框架下不确定性度量研究[J]. 模糊系统与数学, 2020, 34(6): 140-149.